

# 資料結構HW2

四資工二甲41243137陳冠宇

November 25 , 2024

# 目錄

- 1.解題說明
- 2.程式實作
- 3.效能分析
- 4.測試與驗證
- 5.申論及開發報告

# 1.解題說明

- 解題的核心是設計一個多項式類別來進行基本運算（加法、乘法、評估）。首先，定義單項式 (**Term**) 和多項式 (**Polynomial**) 的結構，並通過動態陣列存儲多項式的每一項。需要考慮處理多項式運算時指數排序與合併的邏輯，例如進行加法時合併同次項或乘法時正確地計算係數與指數。

## 2.程式實作

```
int main() {
    Polynomial p1, p2;

    cout << "輸入第一個多項式：" << endl;
    cin >> p1;
    cout << "輸入第二個多項式：" << endl;
    cin >> p2;

    Polynomial sum = p1.Add(p2);
    Polynomial product = p1.Mult(p2);

    cout << "第一個多項式為：" << p1 << endl;
    cout << "第二個多項式為：" << p2 << endl;
    cout << "兩個多項式的和為：" << sum << endl;
    cout << "兩個多項式的積為：" << product << endl;

    float x;
    cout << "請輸入欲求值的 x：" ;
    cin >> x;

    cout << "第一個多項式在 x = " << x << " 的值為：" << p1.Eval(x) << endl;
    cout << "第二個多項式在 x = " << x << " 的值為：" << p2.Eval(x) << endl;

    return 0;
}
```

# 3.效能分析

## 1. 時間複雜度：

- Add:  $O(n+m)$ ，其中  $n, m$  為兩多項式的項數。
  - Mult:  $O(n \cdot m)$ 。
  - Eval:  $O(n)$ 。
- 空間複雜度：動態陣列需求為  $O(n)$ 。

# 4. 測試與驗證

- 測試範例：

1. 測試加法：

輸入  $p(x)=2x^2+3x$ ， $q(x)=x^3+x+1$ 。

預期輸出  $p(x)+q(x)=x^3+2x^2+4x+1$ 。

2. 測試乘法：

輸入  $p(x)=2x^2+1$ ， $q(x)=x+1$ 。

預期輸出  $p(x) \cdot q(x)=2x^3+2x^2+x+1$ 。

3. 測試評估：

$p(x)=x^2+x+1$ ，代入  $x=2$  預期輸出 7。

## 5. 申論及開發報告

- 這個程式主要設計了一個處理多項式運算的架構，透過物件導向的方式建立 **Term** 和 **Polynomial** 類別，清楚地將單項式和多項式的邏輯分開處理。過程中最大的挑戰在於確保動態記憶體管理的正確性，以及處理多項式加法和乘法時，指數相同項目的合併。程式設計還需要考量未來擴展的彈性，例如動態陣列的擴容及保護私有成員的存取。完成這個專案後，我更深刻地理解了 **C++** 中類別設計、記憶體管理及運算子多載的重要性，也體會到細節對於程式正確性和效能的關鍵影響。

END